



**HAL**  
open science

# SINr: a python package to train interpretable word and graph embeddings

Thibault Prouteau, Nicolas Dugué, Simon Guillot, Anthony Perez

## ► To cite this version:

Thibault Prouteau, Nicolas Dugué, Simon Guillot, Anthony Perez. SINr: a python package to train interpretable word and graph embeddings. French Regional Conference on Complex Systems, May 2023, Le Havre, France. pp.215, 10.5281/zenodo.7957531 . hal-04113024

**HAL Id: hal-04113024**

**<https://hal.science/hal-04113024v1>**

Submitted on 1 Jun 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

---

# SINr: a python package to train interpretable word and graph embeddings

Thibault Prouteau · Nicolas Dugué ·  
Simon Guillot · Anthony Perez

**Abstract** In this paper, we introduce the SINr Python package to train word and graph embeddings. The SINr approach is based on community detection: a vector for a node is built upon the distribution of its connections through the communities detected on the graph at hand. Because of this, the algorithm runs very fast, and does not require GPUs to proceed. Furthermore, the dimensions of the embedding space are interpretable, those are based on the communities extracted. The package is distributed under Cecill-2.1 license and is available on Github and pypi.

**Keywords** Graph embedding · Word embedding · Interpretability · Frugal machine learning

## 1 Introducing SINr

With neural approaches, tremendous progress was made in natural language processing, notably to represent the vocabulary of the language at hand, those representations are then used as input for machine learning algorithms. These representations are dense numeric vectors named word embeddings. Some examples of approaches to train such vectors using large textual corpora are Word2vec [6], Glove [8], and the Transformer-based approached for contextualized representations, Camembert [5] or Flaubert [4] in French. This progress was transfered to the graph universe, allowing the emergence of graph embedding, a whole field of research with Word2vec inspired approaches such as Node2vec [3], matrix factorization methods like HOPE [7] and auto-encoding paradigms [2].

---

Thibault, Nicolas and Simon  
Le Mans Université, LIUM, EA 4023, Laboratoire d'Informatique de l'Université du Mans  
E-mail: first.last@univ-lemans.fr

Anthony  
Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans  
E-mail: anthony.perez@univ-orleans.fr

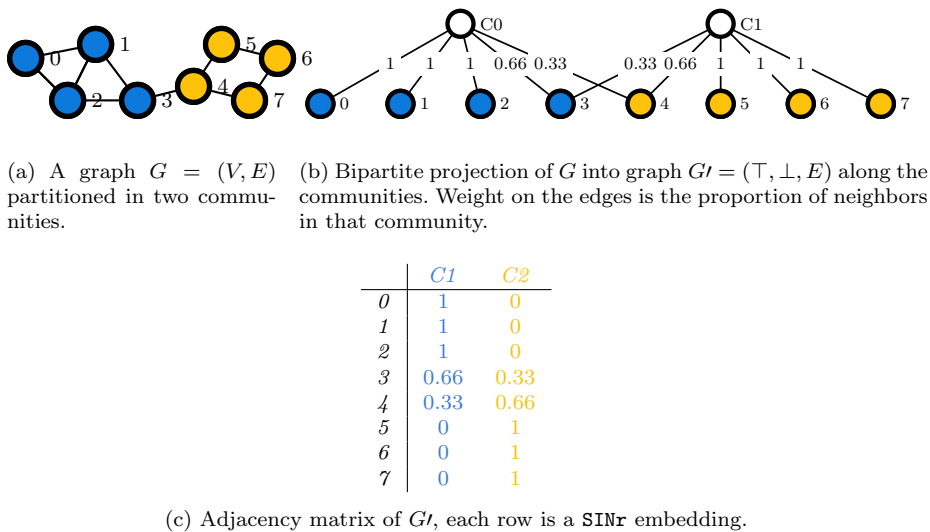


Fig. 1: Illustration of SINr, vertices are represented based on the communities they are linked to.

SINr was introduced to take advantage of this progress, it allows to embed words and nodes just like the aforementioned methods. However, it is based on community detection: for each node, the vector of embedding is calculated as the proportion of its links going to the communities as described Figure 1. This approach allows to avoid some flaws inherent to the usual approaches:

- As far as we know, SINr is the first approach specifically designed to deal with both word and graph embeddings. Textual corpora are represented as graphs, and with the adequate preprocessing provided by the package, word embeddings can easily be extracted with SINr. For graph embedding, no specific pre-processing is required.
- Contrary to the neural approaches that require complex GPU calculations, SINr is based on the Louvain [1] algorithm to detect community and thus runs in linear-time, it can be executed on standalone laptops.
- Contrary to the usual approaches, because dimensions are based on the communities, the space in which words and graphs are embedded with SINr is interpretable.

The performances of SINr were evaluated on several tasks, including link prediction on graphs, and pair of words similarities for textual data [9]. While providing good performances, it runs faster than most of the other embedding approaches. Furthermore, the interpretability of the model was also demonstrated to be comparable to the state-of-the-art when considering word embedding [10]. In this paper, we consider the SINr package that we distribute on <https://github.com/SINr-Embeddings/sinr> and that can also be found on Pypi to be installed with pip.

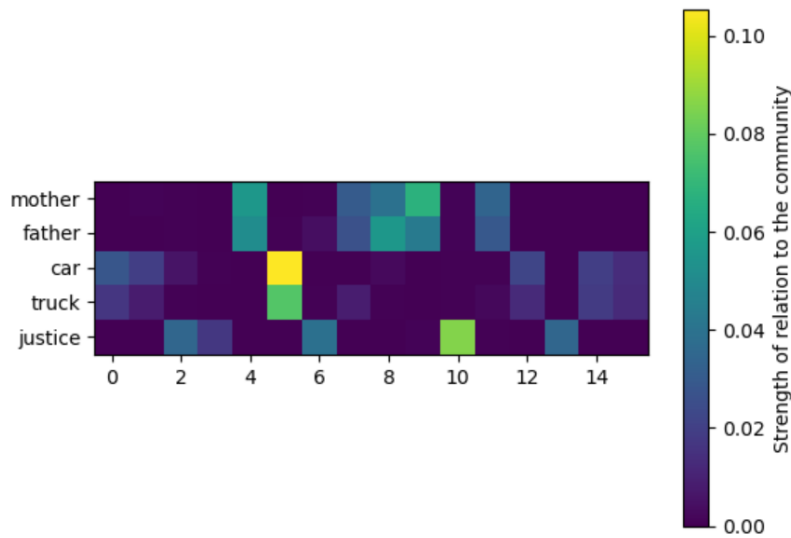


Fig. 2: Because of the sparsity of the embedding space, one can see that related words have non-zero values for the same dimensions (abscissa), *mother* and *father* for dimensions 4, 7, 8, 9 and 11 for instance. The non-zero dimensions are distinct when comparing *mother* and *car* that are unrelated.

## 2 Playing with SINr

One can setup SINr using `pip install sinr`, and train word and graph embeddings using the package features. While these embedding features are shared among models, SINr brings the option to probe and get an understanding of the resulting embedding space as one can see Figure 2. Using SINr leads to sparse vectors: a node is not connected to all the communities of a graph, and similarly, a word is not related to all the topics of a corpus. As shown by [11], sparsity is one of the features required to enforce interpretability. The other one is to embed data in spaces larger than the classic 128 dimensions ones. The  $\gamma$  resolution parameter of the Louvain [1] algorithm allows to vary the number of communities, thus controlling the number of dimensions.

## 3 Conclusion

We presented SINr, a package to train word and graph embeddings based on community detection. It runs in linear-time on standalone laptops and leads to interpretable spaces that can be inspected using features provided in the package. In the future, we plan to deal with temporal graphs and thus temporal embeddings with SINr and to include more features to visualize vectors.

**Acknowledgements** This work was funded by ANR-21-CE23-0010 DIGING.

## References

1. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
2. Shaosheng Cao, Wei Lu, and Qionghai Xu. Deep neural networks for learning graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
3. Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
4. Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*, 2019.
5. Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019.
6. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
7. Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114, 2016.
8. Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
9. Thibault Prouteau, Victor Connes, Nicolas Dugué, Anthony Perez, Jean-Charles Lamirel, Nathalie Camelin, and Sylvain Meignier. Sinr: Fast computing of sparse interpretable node representations is not a sin! In *IDA*, pages 325–337, 2021.
10. Thibault Prouteau, Nicolas Dugué, Nathalie Camelin, and Sylvain Meignier. Are embedding spaces interpretable? results of an intrusion detection evaluation on a large french corpus. In *LREC*, 2022.
11. Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.